

# Simulation and reconstruction of particle trajectories in the CEPC drift chamber\*

Meng-Yao Liu,<sup>1</sup> Wei-Dong Li,<sup>2,3</sup> Xing-Tao Huang,<sup>1,†</sup> Yao Zhang,<sup>2,3</sup> Tao Lin,<sup>2,3</sup> and Ye Yuan<sup>2,3</sup>

<sup>1</sup>*Institute of Frontier and Interdisciplinary Science, Shandong University, 266237 Qingdao, Shandong, China*

<sup>2</sup>*Institute of High Energy Physics, 100049 Beijing, China*

<sup>3</sup>*University of Chinese Academy of Sciences, 100049 Beijing, China*

The Circular Electron Positron Collider (CEPC) is designed to precisely measure the properties of the Higgs boson, study electroweak interactions at the Z-boson peak, and search for new physics beyond the Standard Model. As a component of the 4<sup>th</sup> Conceptual CEPC detector, the drift chamber facilitates the measurement of charged particles. This study implemented a Geant4-based simulation and track reconstruction for the drift chamber. For the simulation, detector construction and response were implemented and added to the CEPC simulation chain. The development of track reconstruction involved track finding using the Combinatorial Kalman Filter method and track fitting using the tool of GenFit. Using the simulated data, the tracking performance was studied. The results showed that both the reconstruction resolution and tracking efficiency satisfied the requirements of the CEPC experiment.

Keywords: CEPC, Drift chamber, Detector simulation, Track reconstruction

## I. INTRODUCTION

In 2012, the discovery of the Higgs boson by the ATLAS [1] and CMS [2] experiments at the Large Hadron Collider (LHC) [3] confirmed the predictions of the Standard Model (SM) [4] with tremendous success. However, the Standard Model does not adequately explain fundamental physical phenomena such as dark matter, dark energy, and neutrino oscillations. With the distinct properties predicted in different theoretical models, the Higgs boson has become a critical piece of the puzzle in the search for physics beyond the Standard Model (BSM) [5, 6]. There are several future collider experiments, including future circular collider (FCC) [7], Circular Electron Positron Collider (CEPC) [8], International Linear Collider (ILC) [9], and Compact Linear Collider (CLIC) [10], which aim to precisely measure the properties of the Higgs boson [11–14]. The CEPC is a 100 km circular electron-positron collider proposed by Chinese physicists. It has been designed to provide electron-positron collisions at a central mass energy of approximately 240 GeV for the Higgs boson study through the  $e^+e^- \rightarrow ZH$  process, as well as collisions at the Z-boson peak for precise electroweak physics measurements.

The abundant physics research programs at CEPC impose stringent requirements on detector performance. The detailed performance requirements can be found in the CEPC Conceptual Design Report (CEPC CDR) [8] published in October 2018. For the tracker, a tracking efficiency of better than 99% is required for charged particles with a transverse momentum greater than 1 GeV, and momentum resolutions of the reconstructed tracks should be achieved per mille level.

The 4<sup>th</sup> conceptual detector is another option for detector design, which was proposed based on the CEPC CDR. The first three detector designs do not include a drift chamber, which is an innovative detector design. Other important

parameters, such as the magnetic field and materials, are consistent with the CEPC CDR and will be updated based on performance testing and future research. As shown in Figure 1, the detector design comprises seven subdetectors. From the innermost to outermost radii, they comprise the VertX Detector (VXD), Silicon Internal Tracker (SIT), Drift Chamber (DC), Silicon External Tracker (SET), Transverse Crystal-bar Electromagnetic Calorimeter (ECAL), Scintillator Glass Hadronic Calorimeter (HCAL), and Muon Tracker. The coil of a 3 Tesla superconducting magnet is located outside the ECAL, and a flux return yoke is embedded in the Muon Tracker. One of the characteristics of this detector is the combination of silicon trackers with a multilayer drift chamber to achieve a higher performance for both tracking and particle identification. For this detector design, the DC functions as a tracking detector that provides an accurate measurement of the position of a charged particle [15–17] and is applied to identify the particle according to the number of ionization clusters lying on the particle track [18].

Both the detector design and potential physics studies require strong support from detector simulation and event reconstruction. Therefore, software development for DC simulation and track reconstruction is critical for CEPC Research and Development (CEPC R&D). This study elucidates the details of the implementation of the DC software and is organized as follows. Section II provides an overview of the CEPC software and the workflow of offline data processing. In Section III, the DC simulation is introduced, including a geometry description[19], detector construction, and digitization [20]. Then, Section IV describes the implementation of track reconstruction, including track finding based on a Combinatorial Kalman Filter (CKF) [21, 22] and track fitting using GenFit [23]. Finally, the physics performance is presented in Section V, followed by a conclusion in Section VI.

## II. SOFTWARE OVERVIEW

CEPC Software (CEPCSW) [24] is an offline software system developed to support data processing and analysis in the

\* This work was supported by the National Natural Science Foundation of China (NSFC) under Contracts No. 12025502 and No. 12341504.

† Corresponding author, [Xingtao.Huang, huangxt@sdu.edu.cn](mailto:Xingtao.Huang, huangxt@sdu.edu.cn)

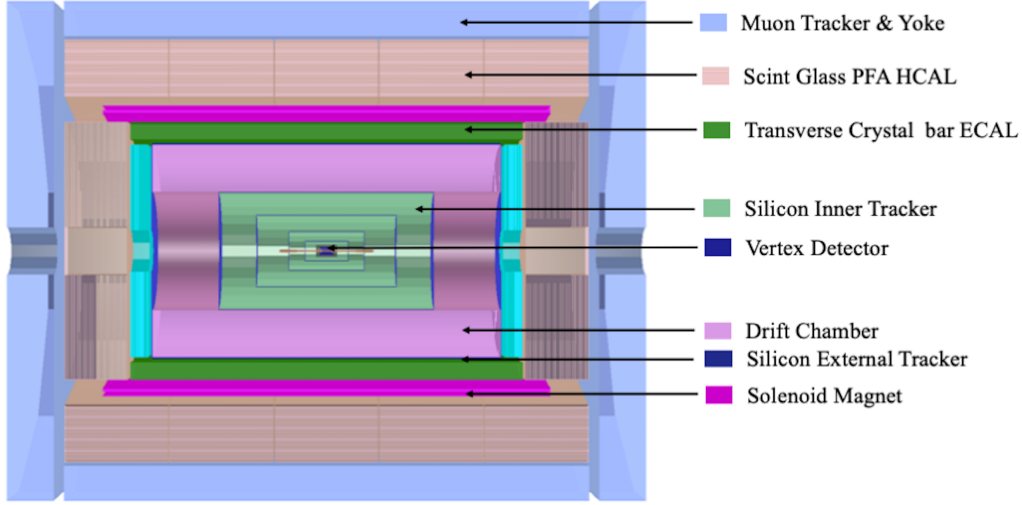


Fig. 1. Schematic of the CEPC 4<sup>th</sup> conceptual detector.

CEPC experiment. The components in the data processing chain include physics generators, Geant4-based detector simulation [25, 26], beam-related background mixing, fast simulation, machine learning interfaces, and event reconstruction. The current software architecture of CEPCSW is illustrated in Figure 2. All software tools and interfaces in the figure were developed to satisfy the requirements of the CEPC program. They are closely related to either detector design or other scientific activities, such as potential physics studies and test-beam data analysis.

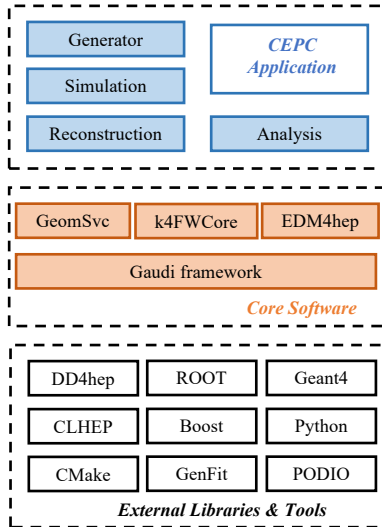


Fig. 2. Architecture of the CEPCSW.

The CEPCSW is fully integrated with Key4hep [27], a

common software stack developed for future high-energy physics (HEP) experiments. The primary parts of the CEPCSW core software include GeomSvc, k4FWCore, EDM4hep [28], and Gaudi [29] framework. GeomSvc is a service through which tracking algorithms can access the detector geometry from DD4hep[30]. Edm4hep functions as an event data model; however, it also provides bidirectional conversion between persistent data and data objects in memory, which are handled by k4FWcore. Gaudi is the underlying software framework responsible for defining the interfaces for all software components, managing data objects in memory, and controlling the execution of applications.

The data processing flow of single-particle events is shown in Figure 3 and comprises four steps. Event generation produces a list of particles, each of which is generated by a single interaction with a vertex located at the geometric origin. In the next step, these generated events are passed into the simulation, where each particle is propagated through the detector using Geant4. During this process, the simulated interactions between the particles and the detector are recorded. In the digitization step, the responses of the elementary detector modules are modelled. In addition to Monte Carlo (MC) hits from signal events, digitization also accepts hits from background events as input. In the final step, the reconstruction reads the charge and/or time information and generates tracks and showers for the tracking detector and calorimeter, respectively.

As mentioned above, EDM4hep is adopted as the event data model, which describes the event data generated at each step as well as the relationships between two successive steps. Additional helper classes are required to facilitate the rapid navigation between a generated event and its corresponding reconstructed event.

The detector is described using DD4hep to ensure that

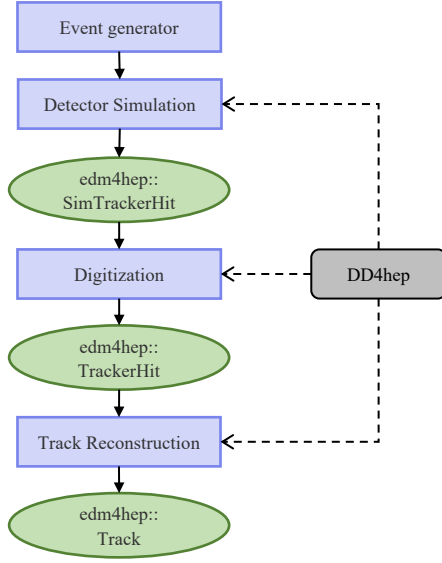


Fig. 3. Data processing flow. The blue rectangles represent data processing algorithms, while the green ellipses define the data objects produced by the last step and used by the followed step.

an identical detector geometry is used for different applications in the workflow. The core element is based on ROOT TGeo[31] with various types of extensions to provide a consistent detector description for simulation, reconstruction, and analysis. DDG4 is used for detector simulation to convert the TGeo-based geometry to the Geant4 geometry, whereas DDRec is employed to provide higher-level detector geometry, for example, information for subdetector components.

### III. DRIFT CHAMBER SIMULATION

The detector simulation relies on the Geant4 simulation toolkit, which provides physics models and realizes particle transportation through geometry. In the CEPSCW, the Geant4 run manager is wrapped with a Gaudi service, facilitating the Gaudi framework in controlling the event loop of the simulation. This service is responsible for initializing the geometry, physics lists, and user actions of Geant4. It also provides standard user interfaces for interacting with Geant4. Because of the simulation service, only the detector geometry and response must be implemented for a specific detector. A precise description of the detector geometry should contain exact knowledge of the position, shape, dimension, and material content of every detector component. The digitization algorithm is invoked when the energy deposited in the sensitive regions of the detector increases above a pre-configured threshold within a particular time window. The digitization output may have provided a detailed signal shape during this period.

#### A. Detector geometry

The DC is surrounded by silicon detectors in both the barrel and end-cap regions, covering the radial range of 800–1,800 mm and the Z range of  $\pm 2,980$  mm. The detailed parameters of the DC are listed in Table 1.

Table 1. Baseline configuration of the drift chamber.

Geometry Parameters	Value
Half length	2980 mm
Inner and outer radius	800 mm ~ 1800 mm
Number of layers	55
Cell size	18 mm $\times$ 18 mm
Gas	90%He+10%C <sub>4</sub> H <sub>10</sub>
Single wire resolution	110 $\mu$ m
Sense to field wire ratio	1:8
Total number of sense wire	25,357
Stereo angle	0.028 rad~0.062 rad
Sense wire	Gold plated Tungsten $\phi = 20 \mu$ m
Field wire	Silver plated Aluminum $\phi = 40 \mu$ m
Wall	Carbon fiber 0.2 mm(inner) and 2.8 mm(outer)

A small cell design is selected to obtain a sufficient number of track hits at the outer radius. The ratio of the half-height to the half-width (the distance between a sense wire and its neighboring field wire in the  $r - \phi$  plane) is approximately 1; thus, the shape of the drift cell is almost square. The drift cell contains a sense wire surrounded by eight field wires. In total, there are 25,357 drift cells of size 18 mm  $\times$  18 mm. To solve the ambiguity [32] caused by multiple hits in the same layer, the drift cells between neighboring layers are offset by the half-width.

Because stereo wires have technical advantages in measuring the positions of charged tracks in the Z-direction, the DC is composed purely of stereo wires. All stereo wires are organized into 55 coaxial layers. The stereo angle on each layer varies as 0.028–0.062 rad, and the wires on any two neighboring layers are tilted in opposite directions.

Materials with low densities and atomic numbers are primarily considered to minimize multiple scattering. Both the inner and outer cylinders are made of carbon fibers. The working gas for the DC is a mixture of helium and C<sub>4</sub>H<sub>10</sub> at a ratio of 90:10. The sense wire is made of gold-plated tungsten with a diameter of 20  $\mu$ m, and the field wire is made of silver-plated aluminum with a diameter of 40  $\mu$ m.

The DC is described using DD4hep, and the detector geometry parameters, such as dimensions, materials, and sensitive detectors, are stored in XML(eXtensible Markup Language)[33] format. XML serves as a compact file for DD4hep and the syntactic structure of the compact XML description is obtained from the SiD[34] detector description. Therefore, the detector design is associated with a set of corresponding XML files. By versioning the changes in the set of XML files, the geometric version can be easily controlled.

In the simulation, the DC is constructed by parsing the geometric parameters from the XML files using a specialized detector constructor, creating individual components, and placing them accurately in the correct positions to form a com-

plete virtual detector. The construction process is as follows. A cylindrical-shaped volume is created according to the maximum and minimum radii and length of the DC. Subsequently, within the cylinder volume, wire layers with thicknesses of 18 mm are created at different radii, each of which is represented by a hollow cylinder. For a specific layer, the sense and field wires are directly constructed as tubes and placed at the right locations. The field wires are shared between neighboring cells with each field wire assigned to one drift cell as an entire tube. Because of the large number of wires, if every drift cell is created independently, it deteriorates the performance in terms of both speed and memory. To solve this problem, an alternative method is implemented to first construct the layers and then divide each layer into drift cells by utilizing DDSegmentation[30] from DD4hep.

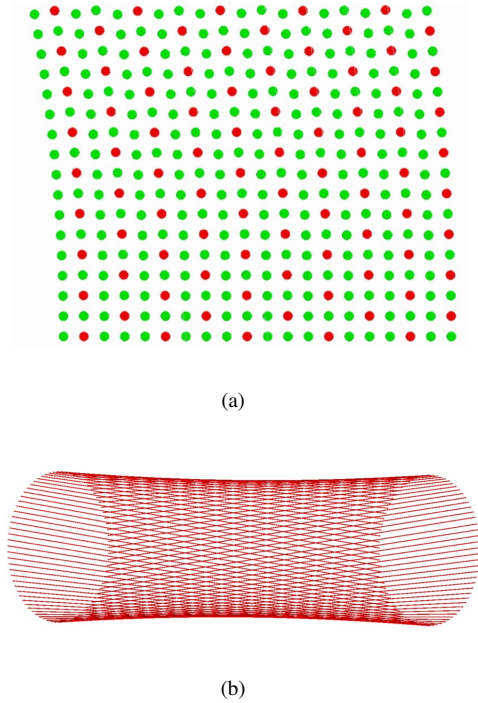


Fig. 4. (a) Projection of the first 10 layers of the DC in the  $r-\phi$  plane. The red dots represent sense wires, while the green dots represent field wires. (b) Visualization of sense wires in the first layer. The skew is exaggerated.

After the geometrical model for the DC is built, the Geant4 visualization tool can be used to produce graphical representations of the detector and draw views and sections of the detector. Figure 4(a) shows a visualization of the sense and field wires created in the simulation, illustrating the  $r-\phi$  projection of the proportion of the first ten layers of wires. The sense wires of each layer forms a rotating hyperboloid surface and Figure 4(b) shows the sense wires in the first layer. With the help of the visualization tool, the parameters of all individual geometrical entities can be displayed and checked. Careful validation of the geometry lays a solid foundation for the next steps in digitization and tracking.

## B. Digitization

The DC is used to measure the spatial coordinates of the trajectory of a charged particle. This is achieved by detecting the ionization electrons produced by the charged particles in the gas of the DC and measuring their drift time and arrival positions on the sense wires.

In the context of detector simulation, digitization refers to the simulation of the detector response. For the DC, an accurate simulation of the drift time is particularly important. Because the CEPC detector R&D is in its early stages, a detailed detector design is still not available. A simplified digitization method is implemented to support the development of the tracking algorithm. Its workflow is illustrated in Figure 5(a).

In the simulation using Geant4, a small step size i.e 0.5 mm) is selected to simulate the passage of charged particles through the DC. When the particle enters the drift cell, the distance between each Geant4 step and the sense wire of the cell is recorded. The smallest distance, referred to as doca (the distance of closest approach), represents the closest approach of the particle trajectory to the sense wire as shown in Figure 5(b), and is considered as the drift distance of this hit. The doca is smeared using a Gaussian function with a width equivalent to the wire resolution set to  $110\ \mu\text{m}$ . The drift time is calculated using the space-time relation (X-T relation) [35]. In this study, a relatively simple linear X-T relationship is used.

During digitization, the effects of various types of background events are considered. The hits from a signal event are overlaid with those from additional background events before the detector response was calculated. The background primarily originates from the minimum bias and beam-gas interactions. In this study, background events are assumed to be randomly distributed within a time window of 2,000 ns. For simplicity, track recognition is only performed on the group of the fastest hits, each of which has the minimum drift time among all hits belonging to a drift cell. Therefore, during the digitization stage, when a background hit precedes that of a signal within the same drift cell, the signal hit is overridden by the background hit. Furthermore, the default wire efficiency is assumed to be 100%, and all subsequent results are based on this assumption.

## IV. TRACK RECONSTRUCTION

Track reconstruction is among the most crucial data processing tasks in HEP experiments because it facilitates high-precision position and momentum measurements and identification of charged particles [36]. A tracking algorithm is developed to implement track reconstruction by combining the spatial measurements (hits) of a traversing particle into sets (tracks) to extract the kinematic properties of the particle. The process of track reconstruction comprises two primary stages: track finding and track fitting, which complement each other to provide the most accurate track parameters.

At present, the track reconstruction for the DC is based on



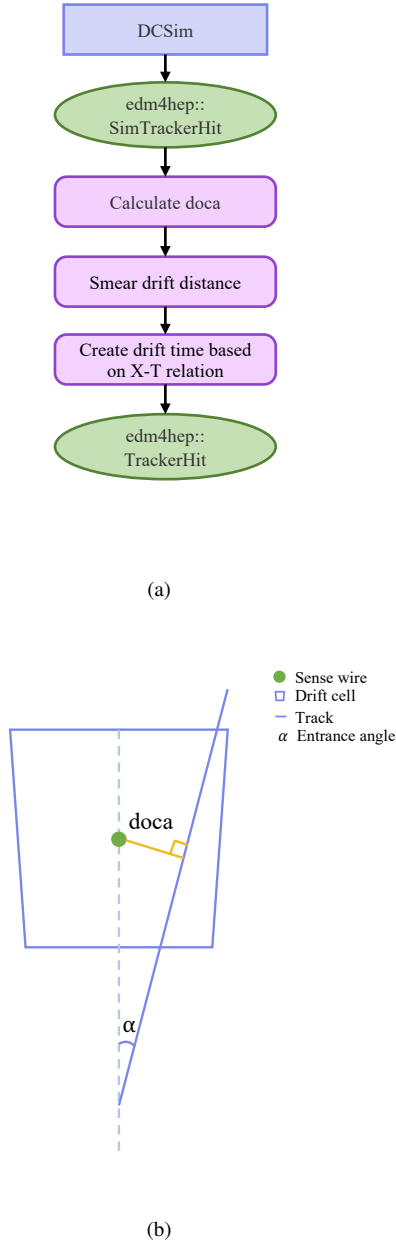


Fig. 5. (a) The specific workflow of digitization. (b) Schematic diagram of the doca.

## A. Track parameterization

In a homogeneous magnetic field, the motion of charged particles in the DC forms a trajectory that is described by the five parameters of a helix, as shown in Equation 1. The first three parameters of the equation describe the circle on the X-Y plane, as shown in Figure 6, and the last two parameters describe the information along the Z direction of the trajectory, following the conventions used by the BaBar experiment [37].

$$\mathbf{P} \equiv (d_0, \phi_0, \omega, z_0, \tan\lambda)^T. \quad (1)$$

The definition of each parameter is as follows:

- $d_0$  is the signed distance from the center of the projected circle of the helix to the reference point on the X-Y plane. Here, the interaction point (IP) is considered as the reference point.
- $\phi_0$  is the azimuth angle of the center of the projected circle of the helix relative to the IP on the X-Y plane.
- $\omega$  is the curvature of the trajectory in the X-Y plane. Further, the sign represents the charge of the trajectory assigned by track fitting.
- $z_0$  is the signed distance between the helix and the IP along the Z direction.
- $\tan\lambda$  is the slope of the helix, that is, the inclination angle of the projection of the helix on the R-Z plane.

Because of material effects and magnetic field inhomogeneity, the real trajectory is not a perfect helix, causing these five parameters to change along the flight path of the trajectory. For the tracking algorithms, all DC hits are considered as measurement data. In this study, charged particles with transverse momentum greater than 0.8 GeV are used. This eliminates the possibility of multiple circular tracks, thus allowing the track to be accurately described by track parameters at the first DC hit. The track parameters are interconverted with the position and momentum of the hits by calling specific functions using the calculation formula referenced in [38].

## B. Extension of geometry interfaces

The track reconstruction process considers the interaction with materials, such as energy loss and multiple scattering, which are highly dependent on the geometry, materials, and nonuniform magnetic field of the DC. To obtain these information, the following extended interfaces are developed by inheriting from GenFit. This because the tracking algorithm is based on the GenFit software package and a detailed explanation of GenFit is presented in Section IV D:

- **CEPCMagneticFieldProvider**: This class inherits from **AbsBField** of GenFit, and provides the magnetic

CKF and it requires seed tracks provided by the silicon trackers. Because silicon trackers can offer high-precision measurements with high efficiency, using silicon tracks as seeds fully meets the requirements raised by the development of the DC track reconstruction algorithm. We plan to develop a dedicated seed-finding algorithm for the DC track reconstruction.

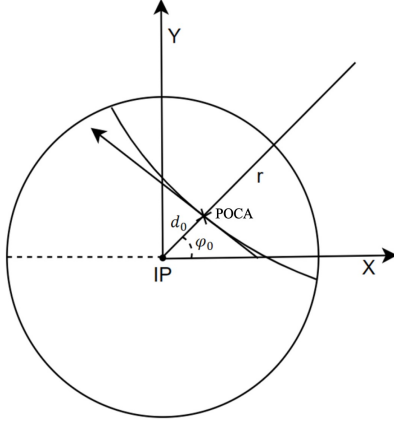


Fig. 6. Description of the helix parameters ( $d_0$ ,  $\phi_0$ ) and the position of the POCA (the point of closest approach).

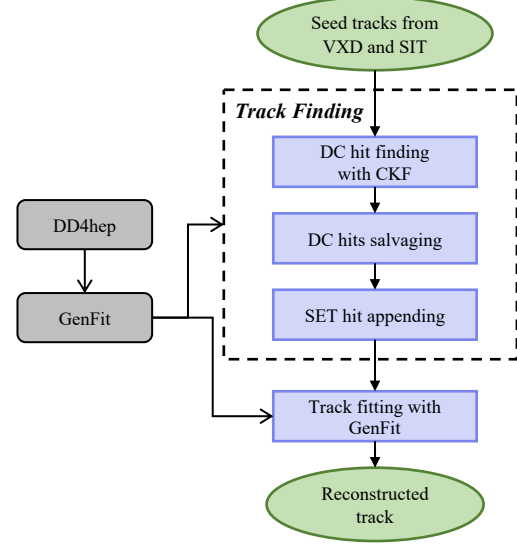


Fig. 7. Workflow of track reconstruction.

field strength at specific positions by integrating with DD4hep. It accepts the position as input and calculates the magnetic field strength from the magnetic field map of the detector.

- **CEPCMaterialProvider:** This class inherits from **AbstractMaterialInterface** of **GenFit**, and obtains the geometry and material information of the DC based on **GeomSvc**.

### C. Track finding

The mission of track-finding is to rapidly and accurately determine the potential DC hits produced by the same charged track based on candidate seed tracks [39]. DC track finding is implemented using three components, as shown in Figure 7: finding DC hits with CKF, salvaging DC hits, and appending SET hits.

#### 1. Hits finding

DC hit finding is among the main procedures in DC track finding, and it is implemented by reusing the CKF algorithm adopted from Belle II [40]. The CKF is an iterative local algorithm, first presented in [41], and widely used in HEP experiments. The CKF starts with a seed estimation of track parameters with uncertainties and then extrapolates the track into the detector volume using the Runge-Kutta-Nyström method [42].

In CEPSCSW, the implementation of the CKF algorithm involves three main tasks: the acquisition of geometry information for the DC, input/output (I/O) management, and optimization of various parameters. The DC geometry can be obtained by implementing classes as introduced in Section IV B. In addition, the I/O of the CKF algorithm exists in the Belle II data format, thus completing the conversion between

EDM4hep and Belle II data model. Because of the different geometrical designs, working gas, and magnetic fields, the judgement criteria and standards of the CKF are optimized, and new thresholds are configured to adapt them to the CEPC DC, for example, the extrapolation track length, chi-squared( $\chi^2$ ) value, and doca.

In this study, five hypotheses are considered: electron, muon, pion, kaon, and proton. A schematic and brief overview of the track-finding process are illustrated in Figure 8. The primary tracking process begins with seed tracks, based on which the search roads are built in the DC. These seeds are extrapolated along the search roads using the CKF, which attempts to smoothen the track while iteratively searching for neighboring hits both outward and inward. After extrapolation, potential candidate hits are identified based on the drift distance, current position, and uncertainties of the candidate track. When multiple candidate hits are found during the propagation step, a prediction is made for each candidate hit using the predicted track parameters and measured values to assess the next candidate hit. Subsequently, a new candidate hit is added to the track and the process is repeated. In cases involving multiple mutually exclusive next-candidate hits, the entire candidate track is duplicated and subsequently treated as two separate tracks. Finally, the final candidate track is selected based on various quality criteria, such as the residual distance between the extrapolated and measured hit positions, extrapolated track length, and the doca, etc.

This process produces a set of potential candidate hits, which are then further refined. The advantage of the CKF algorithm is its fast running speed. However, it is difficult to reliably determine all DC hits because the CKF algorithm uses restrictive conditions, leading to only one hit being found

when multiple hits are present within the same layer. These conditions result in low track quality, particularly for layers with stereo wires and low-momentum particles. Therefore, a hit-salvaging algorithm is developed to solve this problem.

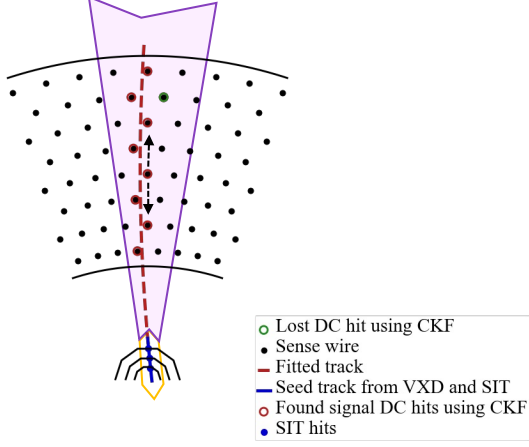


Fig. 8. Schematic of track finding process. For better visibility, the schematic shows only a few hits in the DC. First, determining the DC hits (the red dots) using CKF is based on the seed track (the blue curve), and a rough track (the red curve) is fitted using GenFit. Subsequently, the remaining hits are determined using the hits salvaging algorithm. By extrapolating the track to the remaining hits (the green dots), it is determined whether they belong to the track based on criteria, such as, the doca, residual of the extrapolated distance, etc.

## 2. Hits salvaging and appending

The hit-salvaging algorithm serves as a supplement to the hit-finding algorithm and is developed independently to salvage more signal hits within the DC. This algorithm utilizes the preliminary results from the CKF, uses GenFit to perform rough track fitting, and extrapolates the track to the DC hits lost by the CKF.

During the examination, the hits are evaluated based on selection criteria, including the extrapolated track length and doca, to determine their association with the current track and to add them to the track. Subsequently, by refitting the previous track with the retrieved hits using GenFit, more precise track parameters can be obtained. These two steps can remove the majority of noise hits to achieve high hit purity.

Furthermore, the hit salvaging algorithm facilitates the enhancement of the overall accuracy and completeness of the track reconstruction process, improving track quality and hit efficiency ( $\epsilon_{hit}$ ). The hit efficiency is defined as equation 2:

$$\epsilon_{hit} = \frac{N_{find}}{N_{MC}}, \quad (2)$$

where  $N_{find}$  represents the number of found DC signal hits and  $N_{MC}$  represents the number of truth DC hits. Figure 9

shows the hit efficiency as a function of  $p_T$  measured using single  $e^-$  events, which increased by approximately 26% at 2 GeV.

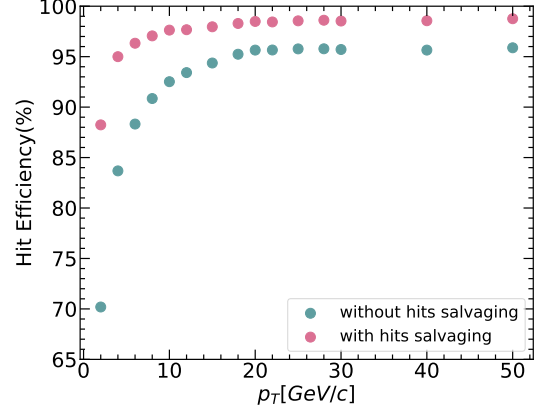


Fig. 9. Hit efficiency as a function of  $p_T$  measured using single  $e^-$ , and the pink dots represent the hit efficiency with the hit salvaging algorithm, while the teal dots represent the hit efficiency without the hit salvaging algorithm.

Next, the refined track is extrapolated to the SET plane using GenFit to determine the expected position of the SET hit. If the extrapolated hit position falls within  $3\sigma$  of the actual hit position on the SET plane, the SET hit obtained by extrapolation is added to the track. This step enhances the accuracy and reliability of the candidate hits.

## D. Track fitting

The track-fitting algorithm is performed to accurately determine the track parameters by fitting all the hits obtained in the aforementioned processes and to provide physics quantities such as momentum, vertex position, and error matrices of the track. Further, this enables the DC to exhibit good track resolution and kinematic discrimination capability [43, 44]. In the track fitting process, the momentum, vertex position, and error matrix of the track are obtained using the method of least squares[45]. The least squares method minimizes the  $\chi^2$  value to determine the parameter estimates for  $\chi^2_{min}$ .  $\chi^2$  is defined as follows.

$$\chi^2 = \sum_{i=1}^n \left( \frac{drift_i - doca_i}{\sigma_i} \right)^2. \quad (3)$$

where  $drift_i$  represents the fitted distance of the  $i^{th}$  hit on the track,  $\sigma_i$  represents the measured error of the drift distance for the  $i^{th}$  hit, and  $doca_i$  represents the distance between the  $i^{th}$  hit sense wire and the fitted track in space, that is, the doca. The fitting process involves multiple iterations to adjust the track parameters until the fit converges.

During the track-fitting procedure, the correct handling of measurement uncertainties and all physical effects disturbing the helical path of the track must be considered to arrive at

a correct description of the track; these effects are highly dependent on the geometry of the detector and the non-uniform magnetic field. Considering all the physical effects renders the mathematical description of the track model very complicated and allows only a numerical calculation. Dedicated fitting algorithms based on GenFit are developed for this purpose because GenFit can accurately determine the track parameters of charged particles using spatial coordinate measurements provided by detectors. The GenFit package is integrated into CEPCSW by extending the necessary interfaces.

### 1. Extension of GenFit interfaces for CEPC

The configuration of GenFit necessitates a large amount of DC information, including the geometry parameters, detector materials, strength distribution of the magnetic field, position of the sense wires, and drift distance of hits, etc. Further, to facilitate the integration of GenFit into CEPCSW, seamless data conversion is necessary. Thus, the following GenFit interface is developed:

- **CEPCToGenFitHitConverter**: Extracting the necessary information based on the TrackerHit object [28] of EDM4hep, and creating a corresponding GenfitHit object which is used as the input of track fitting algorithm. It implements the data model conversion from EDM4hep to GenFit at the level of the hit.
- **CEPCToGenFitTrackConverter**: Storing the results of the track fitting algorithm, including general information (number of iterations, convergence, etc.), fit status ( $\chi^2$ , NDF, p-value, track length, etc.), and constructing a Track object [28] as the output, to achieve data model conversion at the level of the track.
- **WireMeasurementDC**: Inheriting from WireMeasurement [23], which is used for measurements in wire detectors (such as straw tubes and drift chambers). To use this class, a plane must be described by the  $u$  and  $v$  axis with  $v$  coincident with the wire (and  $u$  orthogonal to it, obviously). This is because it is not valid for arbitrary choices of plane orientation.
- **GenfitFitter**: Inheriting from AbsKalmanFitter, which is used to call the fitter in the GenFit package, and enabling the configuration of the magnetic field, material effects, and fitter type.

The interfaces for obtaining material and magnetic field information are introduced in Section IV B, and Figure 10 shows the data flow during the track-fitting process.

After fitting, GenfitTrack objects contain a large amount of data: track representations, TrackPoint objects [23] with measurements, FitterInfo objects [23] that contain all fitter-specific information, and the fitting status used to assess the quality of the fitting. In addition, the fitted drift distance for each hit is obtained, which is useful for subsequent hit selection process. By analyzing this information, the tracks are reconstructed with high track efficiency.

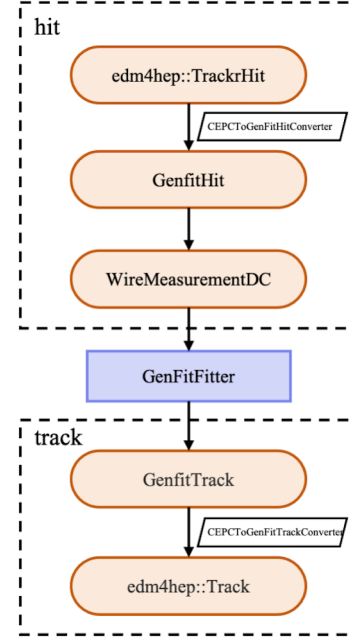


Fig. 10. Data flow of track fitting process.

## V. TRACKING PERFORMANCE

To evaluate the performance of the tracking algorithm, samples including single-particle events and physics process events are generated. Tracks reconstructed using VXD and SIT are used as candidate seed tracks, and the performance of the tracking algorithm is assessed under various conditions.

### A. Data samples

Two types of samples are produced to quantify tracking performance. The first sample encompasses five types of single particles ( $e^-$ ,  $\mu^-$ ,  $\pi^-$ ,  $K^-$ , and  $p$ ) generated using the particle gun of Geant4. The second sample comprises physics events, specifically the  $e^+e^- \rightarrow ZH$ ,  $H \rightarrow \mu^+\mu^-$  process, which is generated using the Whizard [46] generator.

Each single-particle event is generated based on the specified polar angle  $\theta$  and transverse momentum  $p_T$ , whereas the azimuthal angle  $\phi$  is uniformly distributed from  $[-\pi, \pi)$ . The barrel components of SIT and SET provide precise hit positions inside and outside the DC, improving the reconstruction efficiency, particularly for low-momentum charged particles. Therefore, single-particle events with  $\cos\theta < 0.776$  correspond to the barrel region of the DC with a transverse momentum  $p_T$  in the range of [2, 50] GeV.

The  $H \rightarrow \mu^+\mu^-$  process is a clean physics channel with a simple final state, which renders it suitable for testing the algorithm performance. The  $H \rightarrow \mu^+\mu^-$  events are generated by Whizard at leading-order precision, considering the initial state radiation and final state radiation processes during event generation, as well as the 0.16% energy spread of the electron



beam caused by beam-synchrotron radiation in CEPC.

### B. Quality of track fitting

To ensure the reliability of track fitting, it is essential to perform a goodness-of-fit test, which typically includes two components: the  $\chi^2$  test and reduced residuals (pull) test. The  $\chi^2$  value is provided by GenfitTrack object after the fitting. The reduced residuals (pull) are defined as follows:

$$pull = \frac{\nu_{fit} - \nu_{meas}}{[\sigma_{fit}^2 - \sigma_{meas}^2]^{\frac{1}{2}}}, \quad (4)$$

where  $\nu_{fit}$  represents the fitted value obtained from the track reconstruction and  $\nu_{meas}$  is the measured value obtained from the simulation in the previous section. In addition,  $\nu_{fit} - \nu_{meas}$  is the residual between the fitted and measured values, and  $\sigma_{fit}$  and  $\sigma_{meas}$  denote the standard deviations of the fitted and measured values, respectively. The pull distribution conforms to a standard normal distribution with a good fit, indicating that most of the residuals are approximately zero and exhibit a symmetrical shape. The pull test is used to examine track momentum, drift distance, and track parameters. To calculate the pull, the true value is considered with a standard deviation of 0. Consequently, equation (4) is simplified to

$$pull = \frac{P_{fit} - P_{meas}}{\sigma_{fit}}, \quad (5)$$

where  $P_{fit}$  expresses the fitted value and  $P_{meas}$  denotes the measured value.

The momentum resolution is an important metric in track reconstruction. Figure 11 shows the residual distribution between the transverse momentum of the fitted and truth tracks for single  $\mu^-$  with 10 GeV. The distribution closely follows a Gaussian distribution, yielding a momentum resolution of 14.0 MeV, which satisfies the performance requirements of the CEPC tracking system.

Spatial resolution is a critical standard that represents the accuracy of spatial position reconstruction. Figure 12 shows the residual distribution between the fitted and drift distances, which also exhibits a Gaussian distribution. The spatial resolution achieved is 106  $\mu\text{m}$ , which is slightly below the set value of 110  $\mu\text{m}$ . The method of fitting tracks containing measurement points [47], where the construction of  $\chi^2$  in the least-squares fit includes the contribution of the measurement points, can result in an underestimation of the spatial resolution compared to the true value in the reconstruction process.

### C. Track parameters resolution

The main results of the track reconstruction are the five track parameters as well as their distribution and resolution,

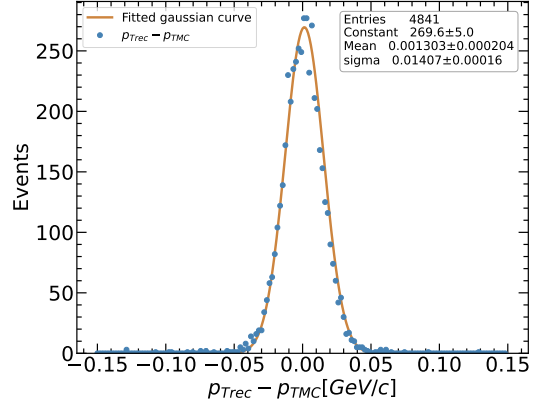


Fig. 11. Blue dots represent the residual distribution between  $p_{Trec}$  and  $p_{TMC}$  for single  $\mu^-$  with 10 GeV.  $p_{Trec}$  and  $p_{TMC}$  represent the transverse momentum of the fitted track and the true track, respectively. The orange line represents the fitted Gaussian function.

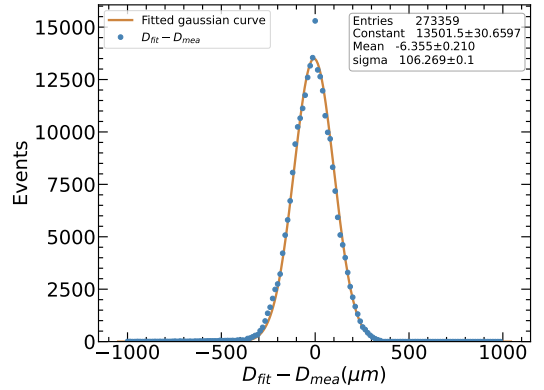


Fig. 12. Blue dots represent the residual distribution between the fitted and drift distances for single  $\mu^-$  with 10 GeV. The orange line represents the fitted Gaussian function.

which are related to the performance of the physics analysis. Owing to the influence of multiple scattering, correlations are observed between the track parameters, with significant correlations among the parameters within the same plane and smaller correlations between the parameters of two different planes. Furthermore, the geometric curvature  $\kappa (= 1/\omega)$  of the track is related to transverse momentum  $p_T$ , where  $\kappa = q/p_T$ . Because the reconstructed instances use the same charge, the resolution of the transverse momentum  $p_T$  of the reconstructed track indirectly reflects the resolution of  $\kappa$ . The following resolutions for  $d_0$ ,  $z_0$ , and transverse momentum  $p_T$  are provided to verify the results of the track reconstruction.

The resolution of the track parameters is obtained by fitting the distribution of the track parameters residuals using a Gaussian function. In the top panels of Figure 13, the resolutions of the impact parameters  $d_0$ ,  $z_0$  and the relative resolution of  $p_T$  are shown as functions of the particle  $p_T$ . The bottom panels of Figure 13 display the resolutions of the impact parameters  $d_0$ ,  $z_0$  at a fixed transverse momentum  $p_T =$

8 GeV and the relative resolution of  $p_T$  as a function of the polar angle  $|\cos\theta|$ . For each  $p_T$  and  $|\cos\theta|$ , a sample of 20k events of single  $\mu^-$  is generated. Owing to the increased material effect, the resolution of  $d_0, z_0$  deteriorates at lower  $p_T$  and larger  $|\cos\theta|$ . The relative resolution of  $p_T$  is dependent on  $p_T$  because the curvature of the track on the X-Y plane is dominated by the multiple-scattering effect. In the high-momentum region, the momentum resolution is dominated by the single-point resolution of the tracks, resulting in worse resolution with an increase in  $p_T$ .

#### D. Tracking efficiency

Another important criterion for evaluating the track reconstruction performance is the tracking efficiency( $\epsilon$ ), which is related to the tracking algorithm, particle type, and particle momentum. Before analyzing track efficiency, it is essential to establish a definition of a good track. A good track is defined as follows: the  $\chi^2$  value is below 400 and the number of DC signal hits on the track is more than six.

The definition of tracking efficiency  $\epsilon$  is

$$\epsilon(\%) = \frac{N_{rec}}{N_{seed}}, \quad (6)$$

where  $N_{seed}$  is the number of satisfactory seed tracks successfully reconstructed using VXD and SIT, and  $N_{rec}$  is the number of reconstructed good-tracks.

As material effects easily lead to the generation of secondary particles, resulting in inconsistencies between the reconstructed track using MC particles as the seed and true tracks, high-quality track events are selected. When a charged particle enters the DC with minimal energy loss, it is considered to be a high-quality track, which is evaluated by  $(p_{mc} - p_{first})/p_{mc}$ , where  $p_{mc}$  represents the true momentum and  $p_{first}$  represents the momentum of the first DC hit.

Figure 14 shows the track efficiency as a function of  $p_T$  for the five types of single particles, and the track efficiency for all types particles remains consistently above 99.5% without background, particularly for single  $\mu^-$ , and the track efficiency is approximately 100%. Further, when a particle goes through the DC, only the energy loss of the particle is considered. If other interactions occur in the DC material and new types of particles are generated, these events are not considered in the tracking efficiency calculations. Thus, no distinction is made between particles and antiparticles in this study. The evaluation of track reconstruction with the background is important, as the track reconstruction of a real detector is susceptible to the background from different sources, significantly impacting the accuracy of the reconstruction result. To assess the performance of the tracking algorithm with the background and measure its ability to remove noise, the tracking efficiency is tested at different background levels. Figure 15 presents the track efficiency as a function of  $p_T$  for single  $\mu^-$ , both without background and with a noise level of 20%. As evident, track efficiency is not significantly affected by the background, remaining consistently

above 99.8%. This indicated the good stability and robustness of the tracking algorithm. The track multiplicity is a critical and significant issue, and preliminary research has been conducted. The generated MC events exhibit an average of 10 tracks with an average momentum of 2 GeV with a tracking efficiency of 99.6%. This suggests that the current track reconstruction algorithm can handle multiple tracks effectively and achieve good performance.

#### E. Physics event reconstruction

To verify the accuracy of the tracking algorithm in physics events,  $e^+e^- \rightarrow ZH, H \rightarrow \mu^+\mu^-$  process is analyzed. Figure 16 shows the reconstructed mass distribution of the Higgs boson fitted to the CrystalBall function. The mean value is measured at 125 GeV and aligned with the expected invariant mass of the Higgs boson. Further, the standard deviation is measured at 212.5 MeV.

Based on the aforementioned results and performance evaluations, it can be concluded that the tracking algorithm operates effectively and satisfies the performance requirements of the CEPC project.

## VI. CONCLUSION

This study proposes the design and implementation of the DC software for CEPC, comprising four building blocks: event generation, DC simulation, digitization, and track reconstruction. The software is developed using C++ and Python based on the Gaudi framework and relies on several external libraries, including Geant4, CLHEP, ROOT, BOOST, GenFit, EDM4hep, DD4hep, and Key4hep. A modular design style is adopted based on Gaudi dynamically loadable elements, which enhances the maintainability, extensibility, and reliability.

In the DC simulation, the DC geometry is constructed, including layers with stereo wires, materials, and magnetic field, and DDSegmentation is used to divide the virtual drift cells, thereby reducing runtime and saving computational resources. Digitization is implemented to simulate the electronic readout, considering a simple uniformly random background. This renders the simulation closer to real data, and provides a large amount of simulated data for reconstruction and physics research.

The tracking algorithm is implemented using four main processes: finding DC hits based on the CKF, salvaging DC hits lost by the CKF, appending the SET hit, and fitting the track using GenFit. First, DC hits are determined using CKF based on the seed tracks provided by the silicon detector. The CKF from Belle II is reused with the conversion of the data model, expansion of the geometry interface, and optimization of the configuration parameters, resulting in the overall best tracking performance. Second, an algorithm for salvaging DC hits is independently developed using GenFit for extrapolation, serving as a supplement to the hit-finding algorithm,

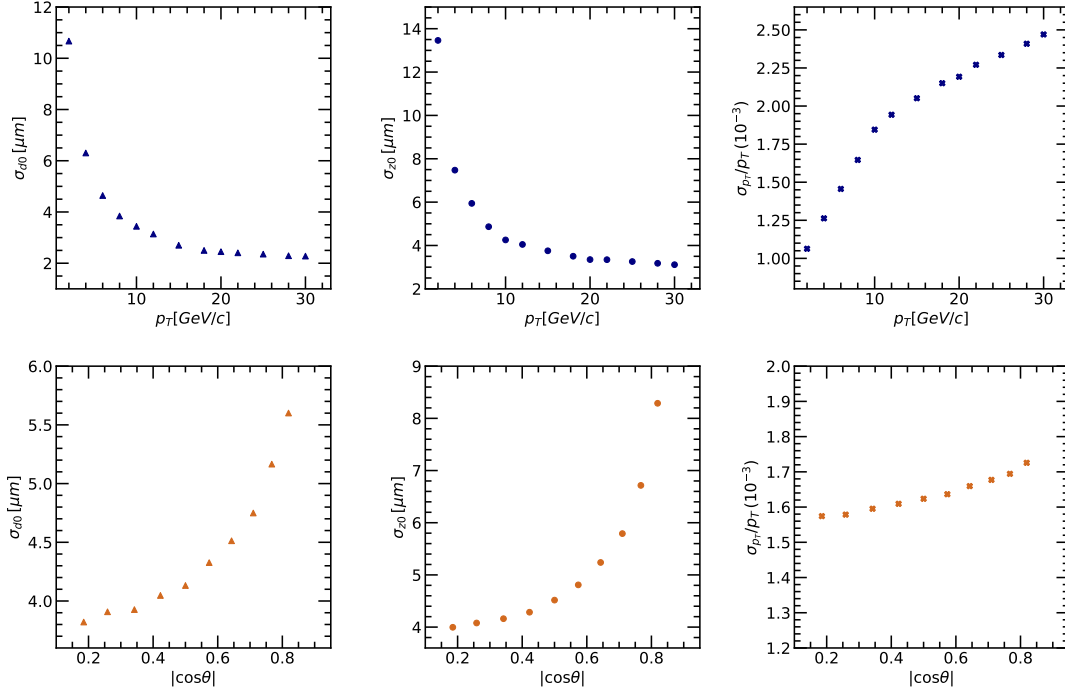


Fig. 13. Resolution of  $d_0$  (left panels),  $z_0$  (middle panels) and relative resolution of  $p_T$  (right panels) for single  $\mu^-$  as function of particle  $p_T$  (top panels) and polar angle  $|\cos\theta|$  (bottom panels) respectively.

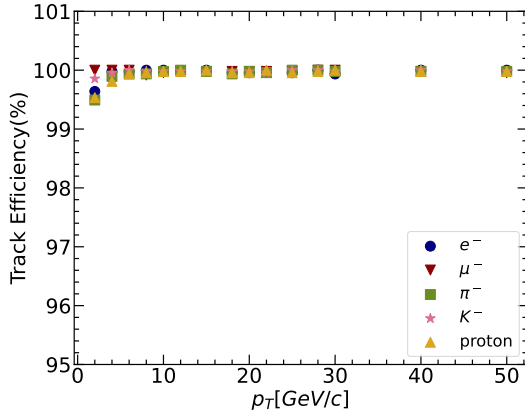


Fig. 14. Tracking efficiency as a function of  $p_T$  measured using single particles, including  $e^-$ ,  $\mu^-$ ,  $\pi^-$ ,  $K^-$ ,  $p$ , without background.

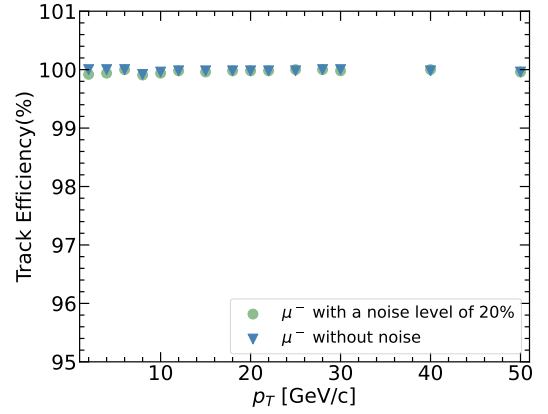


Fig. 15. Track efficiency for single  $\mu^-$  as a function of particle  $p_T$ . The light blue dots represent the results without background, and the light green dots represent the results with a noise level of 20%. For each  $p_T$ , a sample of 5k events is used for the study.

thereby significantly improving hit efficiency and track quality. A SET hit is then added to the track to improve the accuracy of the tracking algorithm. Finally, the tracking toolkit GenFit for track fitting is adopted at CEPC based on a 55-layers drift chamber, and for the first time, a detailed introduction and application in high-energy physics experiments are demonstrated.

The performance of the tracking algorithm has been studied in addition to Geant4-based full simulation of particle interactions with the detectors and  $H \rightarrow \mu^+\mu^-$  events. The fitted tracks are successfully fitted and converged, with a fitting success rate of 99.8%. The time required to reconstruct

a single  $\mu^-$  event with 2 GeV, including track finding and track fitting, has an average of 0.294 s. This number is reasonable. The tracking efficiency is above 99.55% for five types of single particles ( $e^-$ ,  $\mu^-$ ,  $\pi^-$ ,  $K^-$ ,  $p$ ) and for single  $\mu^-$  remains above 99.80% with a noise level of 20%, which is consistent with the case without background. The track efficiency of 99.6% for multiple tracks events generated by 2 GeV  $\mu^-$ , illustrates the track reconstruction algorithm can handle multi-tracks and exhibits good performance. When the momentum is 10 GeV and  $|\cos\theta| = 0.09$ ,  $\sigma_{p_T}/p_T$  is less

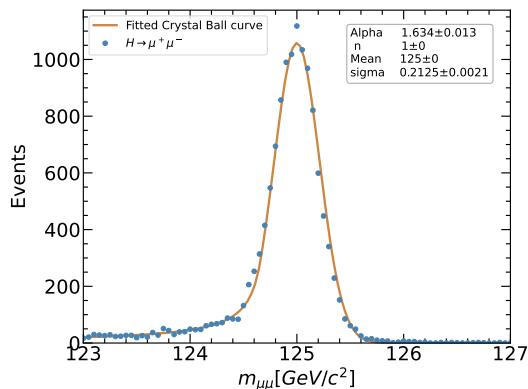


Fig. 16. The reconstructed dimuon mass distributions from the  $H \rightarrow \mu^+ \mu^-$  events.

than 0.14% which satisfies the performance requirements of CEPC. The reconstructed dimuon mass distributions from the  $H \rightarrow \mu^+ \mu^-$  events are reasonable, with a sigma of 0.21 GeV. The results demonstrate that the tracking algorithm exhibits an excellent performance, satisfies the performance requirements of the CEPC tracking system, and can be utilized for further research.

Further development will be based on more realistic simulation of detector response in the DC, and focus on the tracking performance considering wire efficiency and complex beam-induced backgrounds. Moreover, further optimization is required to improve the execution speed of the track reconstruction algorithm. Furthermore, a seed-finding algorithm will be developed to provide seed tracks for DC track reconstruction.

- [1] G. Aad, T. Abajyan, B. Abbott, and et al (ATLAS), Phys. Lett. B **716**, 1 (2012), comments: 24 pages plus author list (38 pages total), 12 figures, 7 tables, revised author list, 1207.7214, URL <https://cds.cern.ch/record/1471031>.
- [2] S. Chatrchyan, V. Khachatryan, A. Sirunyan, and et al, Physics Letters B **716**, 30 (2012), ISSN 0370-2693, URL <https://www.sciencedirect.com/science/article/pii/S0370269312008581>.
- [3] L. Evans and P. Bryant, Journal of Instrumentation **3**, S08001 (2008), URL <https://dx.doi.org/10.1088/1748-0221/3/08/S08001>.
- [4] S. F. Novaes, *Standard model: An introduction* (2000), hep-ph/0001283.
- [5] T. A. collaboration, Tech. Rep., CERN, Geneva (2013), URL <https://cds.cern.ch/record/1611190>.
- [6] H. Cheng, W. H. Chiu, Y. Fang, Y. Gao, J. Gu, G. Li, T. Li, Z. Liang, B. Liu, J. Liu, et al., *The physics potential of the cepec. prepared for the us snowmass community planning exercise (snowmass 2021)* (2022), 2205.08553.
- [7] M. Koratzinos et al., in *Proc. of International Particle Accelerator Conference (IPAC'17), Copenhagen, Denmark, 19 May, 2017* (JACoW, Geneva, Switzerland, 2017), no. 8 in International Particle Accelerator Conference, pp. 3003–3006, ISBN 978-3-95450-182-3, <https://doi.org/10.18429/JACoW-IPAC2017-WEPIK034>, URL <http://jacow.org/ipac2017/papers/wepik034.pdf>.
- [8] T. C. S. Group, *Cepc conceptual design report: Volume 2 - physics & detector* (2018), 1811.10545.
- [9] A. Djouadi, J. Lykken, K. Mönig, Y. Okada, M. Oreglia, and S. Yamashita, *International linear collider reference design report volume 2: Physics at the ilc* (2007), 0709.1893.
- [10] CERN, *Cern yellow reports: Monographs, vol2* (2018): *The compact linear e+e- collider (cllc) : 2018 summary report* (1970), URL <https://e-publishing.cern.ch/index.php/CYRM/issue/view/66>.
- [11] Z.-X. Chen, Y. Yang, M.-Q. Ruan, D.-Y. Wang, G. Li, S. Jin, and Y. Ban, Chinese Physics C **41**, 023003 (2017), ISSN 1674-1137, URL <https://dx.doi.org/10.1088/1674-1137/41/2/023003>.
- [12] T. Han, Z. Liu, and J. Sayre, Phys. Rev. D **89**, 113006 (2014), URL <https://link.aps.org/doi/10.1103/PhysRevD.89.113006>.
- [13] F. An, Y. Bai, C. Chen, and et al, Chinese Physics C **43**, 043002 (2019), ISSN 2058-6132, URL <http://dx.doi.org/10.1088/1674-1137/43/4/043002>.
- [14] Q. Sha, A. Fadol, F. Guo, G. Li, Y. Fang, J. Gu, and X. Lou, The European Physical Journal C **82** (2022), ISSN 1434-6052, URL <http://dx.doi.org/10.1140/epjc/s10052-022-10926-5>.
- [15] D. Ming-Yi, Z. Qin, X. Ma, J. Zhang, J.-Q. Dong, W. Xie, Q. Xiu, X. Ju, and R. Liu, International Journal of Modern Physics: Conference Series **31** (2014).
- [16] L.-M. Lyu, H. Yi, L.-M. Duan, Z. Zhang, and Z.-G. Xiao, Nuclear Science and Techniques **31**, 11 (2020), ISSN 2210-3147, URL <https://doi.org/10.1007/s41365-019-0716-x>.
- [17] S.-W. Tang, S.-T. Wang, L.-M. Duan, Z.-Y. Sun, Y.-H. Yu, X.-H. Zhang, K. Yue, C.-G. Lu, H.-R. Yang, P. Ma, et al., Nuclear Science and Techniques **28**, 68 (2017), ISSN 2210-3147, URL <https://doi.org/10.1007/s41365-017-0217-8>.
- [18] F. Cuna, N. D. Filippis, F. Grancagnolo, and G. F. Tassielli, *Simulation of particle identification with the cluster counting technique* (2021), 2105.07064.
- [19] K.-X. Huang, Z.-J. Li, Z. Qian, J. Zhu, H.-Y. Li, Y.-M. Zhang, S.-S. Sun, and Z.-Y. You, Nuclear Science and Techniques **33**, 142 (2022), ISSN 2210-3147, URL <https://doi.org/10.1007/s41365-022-01133-8>.
- [20] J.-Y. Zhao, N.-N. Miao, L.-H. Wu, L.-L. Wang, T. Sun, Z. Huang, R. Mitchell, W.-G. Li, H.-M. Liu, X.-C. Lou, et al., Radiation Detection Technology and Methods **4**, 174 (2020), URL <https://doi.org/10.1007/s41605-020-00166-0>.
- [21] N. Braun, Springer Theses (2019), URL <https://api.semanticscholar.org/CorpusID:201138327>.
- [22] X. Ai, C. Allaire, N. Calace, A. Czirkos, I. Ene, M. Elsing, R. Farkas, L.-G. Gagnon, R. Garg, P. Gessinger, et al., *A common tracking software project* (2021), 2106.13593.
- [23] J. Rauch and T. Schlüter, Journal of Physics: Conference Series **608**, 012042 (2015), URL <https://doi.org/10.1088/1742-6596/608/2/012042>.
- [24] CEPCSW, *Cepc offline software prototype based on key4hep* (2023), <https://github.com/cepc/CEPCSW/>.
- [25] T. Lin, Y. Hu, M. Yu, H. Zhang, S. C. Blyth, Y. Wang, H. Lu, C. Jollet, J. P. A. M. de André, Z. Deng,



- et al., The European Physical Journal C **83** (2023), ISSN 1434-6052, URL <http://dx.doi.org/10.1140/epjc/s10052-023-11514-x>.
- [26] X.-F. Huang, H.-B. Liu, J. Zhang, B. Huang, W.-J. Xie, H.-B. Feng, X.-C. Cai, X.-W. Liu, Z.-L. Li, J.-Y. Gu, et al., Nuclear Science and Techniques **32**, 67 (2021), ISSN 2210-3147, URL <https://doi.org/10.1007/s41365-021-00903-0>.
- [27] V. Völkl et al., PoS ICHEP2022, 234 (2022).
- [28] F. Gaede, T. Madlener, P. D. Fernandez, and et al, Proceedings of 41st International Conference on High Energy physics — PoS(ICHEP2022) (2022), URL <https://api.semanticscholar.org/CorpusID:254091554>.
- [29] G. Barrand, I. Belyaev, P. Binko, and et al, Computer Physics Communications **140**, 45 (2001), ISSN 0010-4655, cHEP2000, URL <https://www.sciencedirect.com/science/article/pii/S0010465501002545>.
- [30] M. Frank, F. Gaede, C. Grefe, and P. Mato, J. Phys. Conf. Ser. **513**, 022010 (2014).
- [31] H.-K. Wu and C. Li, Nuclear Science and Techniques **32**, 115 (2021), ISSN 2210-3147, URL <https://doi.org/10.1007/s41365-021-00952-5>.
- [32] X. Ai, X. Huang, and Y. Liu, Journal of Instrumentation **18**, P07026 (2023), URL <https://dx.doi.org/10.1088/1748-0221/18/07/P07026>.
- [33] Wikipedia, *The introduction of the xml file* (2023), <https://en.wikipedia.org/wiki/XML>.
- [34] H. Aihara, P. Burrows, M. Oreglia, E. L. Berger, V. Guarino, J. Repond, H. Weerts, L. Xia, J. Zhang, Q. Zhang, et al. (2009), 0911.0006.
- [35] X.-L. Kang, L.-H. Wu, Z. Wu, and et al, Chinese Physics C **39**, 026002 (2015), URL <https://dx.doi.org/10.1088/1674-1137/39/2/026002>.
- [36] L. Qiu-Guang, Z. Shi-Lei, L. Wei-Guo, and et al, Chinese Physics C **32**, 565 (2008), URL <https://dx.doi.org/10.1088/1674-1137/32/7/011>.
- [37] D. N. Brown, E. Charles, and D. A. Roberts (2000), URL <https://api.semanticscholar.org/CorpusID:85519736>.
- [38] J. Lukui, M. Zepu, L. Weidong, and et al, Chinese Physics C **34**, 1866 (2010).
- [39] Q. Xiu, M. Dong, W. Li, H. Liu, Q. Ma, Q. Ouyang, Z. Qin, L. Wang, L. Wu, Y. Yuan, et al., *Study of the tracking method and expected performance of the silicon pixel inner tracker applied in besiii* (2015), 1510.08558.
- [40] V. Bertacchi, T. Bilka, N. Braun, and et al, Computer Physics Communications **259**, 107610 (2021), ISSN 0010-4655, URL <https://www.sciencedirect.com/science/article/pii/S0010465520302861>.
- [41] P. Billoir, Computer Physics Communications **57**, 390 (1989), ISSN 0010-4655, URL <https://www.sciencedirect.com/science/article/pii/001046558990249X>.
- [42] E. Lund, L. Bugge, I. Gavrilenko, and A. Strandlie, Journal of Instrumentation **4**, P04001 (2009), URL <https://dx.doi.org/10.1088/1748-0221/4/04/P04001>.
- [43] W. Ji-Ke, M. Ze-Pu, B. Jian-Ming, and et al, Chinese Physics C **33**, 870 (2009), URL <https://dx.doi.org/10.1088/1674-1137/33/10/010>.
- [44] Y. Zhang, Y. Nakatsugawa, H. Li, Y. Yuan, and T. Xing, EPJ Web Conf. **214**, 02005 (2019).
- [45] H.-H. Song, Y.-G. Yuan, T.-P. Peng, Y. Yang, Y.-G. Mo, and J. Wu, Nuclear Science and Techniques **29**, 118 (2018), ISSN 2210-3147, URL <https://doi.org/10.1007/s41365-018-0454-5>.
- [46] W. Kilian, T. Ohl, and J. Reuter, The European Physical Journal C **71** (2011), ISSN 1434-6052, URL <http://dx.doi.org/10.1140/epjc/s10052-011-1742-y>.
- [47] R. K. Kutschke and A. Ryd (1996).